Question 1 [/2 Points] [0.5 point for each choice]

Choose the correct value for $t(n)$ for each segment of algorithm given.

```
int n = 30000
for (i = 1; i < n; i++)
    print("***")
for (i = 1; i < n/5; i++)
    print("***")
```

A. O(1)	B. O(n)	C. O(n^2)	D. O($n \log n$)

```
for (i = 1; i < n; i++)
    print("****")
for (j = 1; j < n; j++)
    for (k = 1; k < n; k*=2)
        print("****")
```

A. O(1)	B. O(n^2)	C. O(n^3)	D. O($n \log n$)

```
if x < y
    for (i = 1; i < n; i++)
        print("***")
else if x = y
        for (j = 1; j < n; j++)
```

```
for (i = 1; i < n^3; i++)
    print("***")
for (j = 1; j < n^2; j++)
    procedure x(j)
```

A. O(1)	B. O(n)	C. O(n^2)	D. O($n \log n$)

```
if x < y
    for (i = 1; i < n; i++)
        print("***")
else if x = y
        for (j = 1; j < n; j++)
            for (k = 1; k < j; k++)
                print("***")
    else
        for (i = 1; i < n; i*=6)
            print("***")
```

A. O(1)	B. O(n^2)	C. O($n^2 \log n$)	D. O($n^3 \log n$)

A. O(1)	B. O(n^2)	C. O(n^3)

```
for (i = 1; i < n^3; i++)
    print("***")
for (j = 1; j < n^2; j++)
    procedure x(j)


procedure x(int x) {
    for (i = x; i > 1; i--)
        print("***")
}
```

الجواب $O(n^4)$

A. O(1)	B. O(n^n)	C. O(n^3)

Question2 [/ 0.75 Point] [0.25 point per correct answer, O and Ω are also acceptable.]

	C. $O(n^3)$	D. $O(n^4)$

Question2 [____ / 0.75 Point] [0.25 point per correct answer, O and Ω are also acceptable.]

Applying exhaustive search to the three problems listed below, what is the time complexity: n^2

	Problem	Time Complexity
1	The traveling salesman problem	$\Theta((n-1)!)$
2	The knapsack problem	$\Theta(2^n)$
3	The assignment problem	$\Theta(n!)$

OR $O(n!)$ OR $O(\frac{n}{2}!)$ OR $O(\frac{n-1}{2}!)$

Consider the following recursive definition of a recursive algorithm:

$$F(n) = F(n-2) * 2 \quad \text{for } n > 1$$

$$F(0) = F(1) = 2$$

A) What is the basic operation performed? 0.25 Multiplication by 2.

B) **Set up** and **solve** a recurrence relation for the number of times the algorithm's basic operation

For the number of multiplications made by this algorithm

0.25 0.25

$$M(n) = M(n-2) + 1 \quad \text{for } n > 1$$

0.25 (-0.1 if one of them is not written)

$$M(0) = M(1) = 0$$

To solving this recurrence, we use the method of backward substitutions:

0.25 for using the idea of substituting

$$M(n) = M(n-2) + 1$$

substitute $M(n-2) = M(n-4) + 1$